Assignment 3
CS377P Fall 2022
Matthew Jagen (mtj595)
10/11/22

Here is the link to the jupyter notebook I wrote my code in (graphs generated with matplotlib)

1.
a)

a) $x_1=0 \quad y_1=0$

$3x+y=4$

$x+2y=3$

$$\boxed{\begin{aligned} x_{i+1} &= \frac{(4-y_i)}{3} \\ y_{i+1} &= \frac{(3-x_i)}{2} \end{aligned}}$$

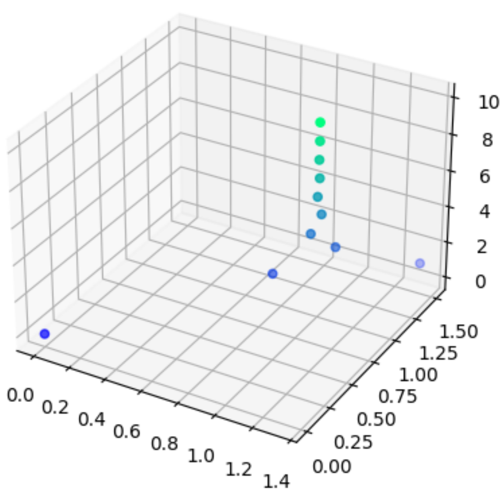
in terms of residual error:

$$\boxed{\begin{aligned} x_{i+1} &= x_i - \tfrac{1}{3}(3x_i + y_i - 4) \\ y_{i+1} &= y_i - \tfrac{1}{2}(x_i + 2y_i - 3) \end{aligned}}$$

in matrix terms:

$$\boxed{\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \end{bmatrix} - \begin{bmatrix} 1/3 & 0 \\ 0 & 1/2 \end{bmatrix} \begin{bmatrix} 3x_i + y_i - 4 \\ x_i + 2y_i - 3 \end{bmatrix}}$$

b)



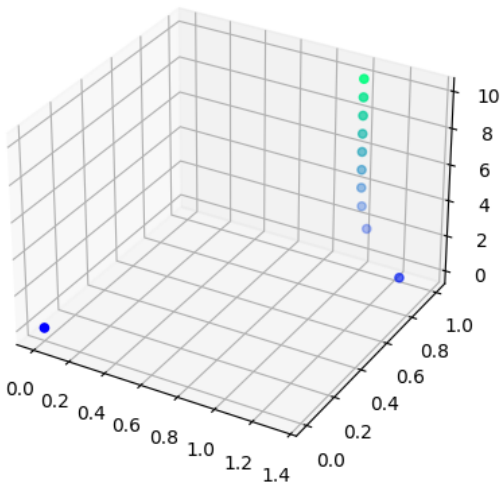
My first intuition when looking at this plot is that our original guess of (0,0) was quite low so its next guess seems to overshot the point where it eventually settles down around. After about 5 data points or so it seems to settle down around the values (1,1)

c)

$$x_{i+1} = \frac{(4-y_i)}{3}$$
$$y_{i+1} = \frac{(3-x_{i+1})}{2}$$

or in matrix terms:

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \end{bmatrix} - \begin{bmatrix} 1/3 & 0 \\ 0 & 1/2 \end{bmatrix} \begin{bmatrix} 3x_i + y_i - 4 \\ x_{i+1} + 2y_i - 3 \end{bmatrix}$$



This graph reaches the same point as the one in part b) of (1,1), but it gets close to it much more quickly. After around 3 points or so it already seems to settle down. Additionally, the second guess overshoots by a much smaller amount than the second guess with jacobi.

2.

a)

$$y''(x) \rightarrow \frac{\frac{y_b(nh)-y_b(nh-h)}{h} - \frac{y_b(nh-h)-y_b(nh-2h)}{h}}{h}$$

$$\left(-y_b(nh) = \frac{y_b(nh)-2y_b(nh-h)+y_b(nh-2h)}{h^2} \right) \cdot h^2$$

$$-h^2 y_b(nh) = y_b(nh)-2y_b(nh-h)+y_b(nh-2h)$$

$$-h^2 y_b(nh) - y_b(nh) = -2y_b(nh-h)+y_b(nh-2h)$$

$$y_b(nh)(-h^2-1) = -2y_b(nh-h)+y_b(nh-2h)$$

$$y_b(nh) = \frac{-2y_b(nh-h)+y_b(nh-2h)}{(-h^2-1)} \quad \cdot \frac{-1}{-1}$$

$$\boxed{y_b(nh) = \frac{2y_b(nh-h)-y_b(nh-2h)}{(h^2+1)}}$$

b)

$$y'(x) \rightarrow \frac{y_b(nh)-y_b(nh-h)}{h}$$

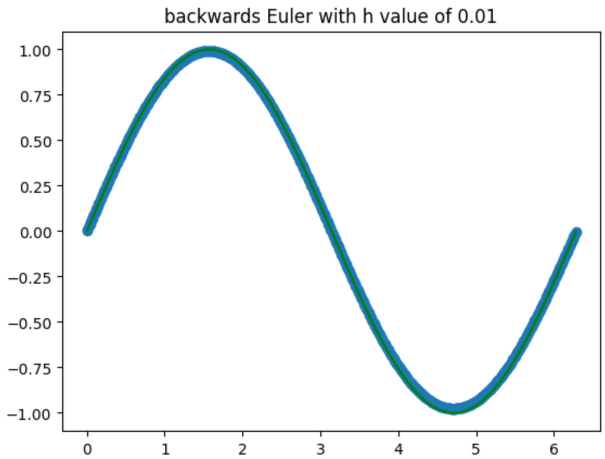
b)

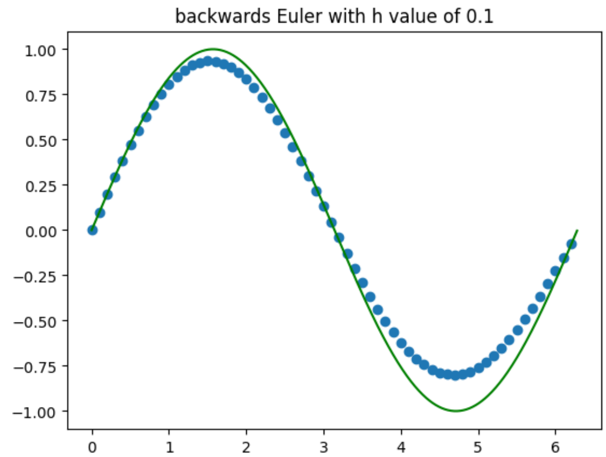
$$y(0)=0 \rightarrow \boxed{y_b(0)=0}$$

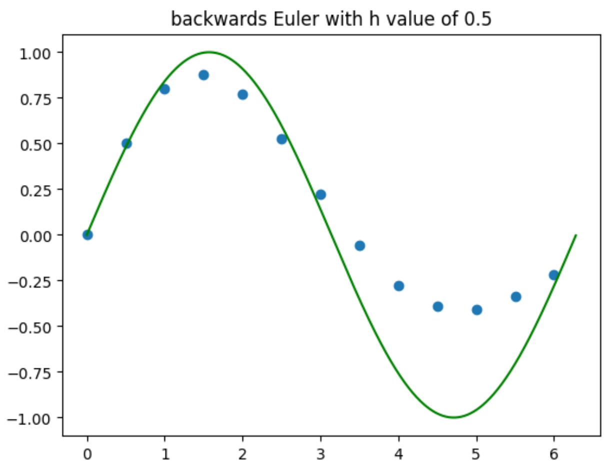
$$y'(0)=1 \rightarrow \frac{y_b(h)-y_b(0)}{h} = 1$$

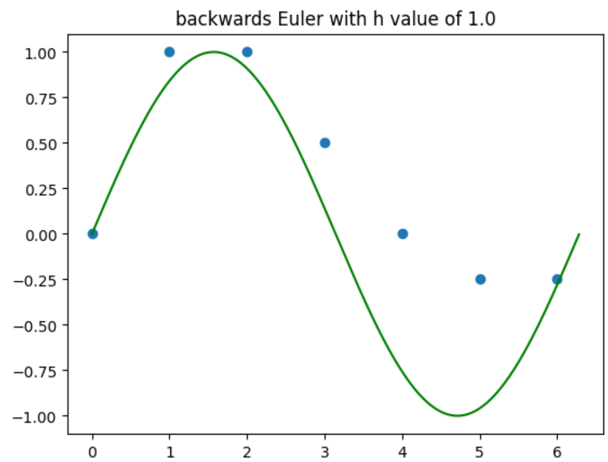
$$\frac{y_b(h)-0}{h} = 1 \qquad \boxed{y_b(h)=h}$$

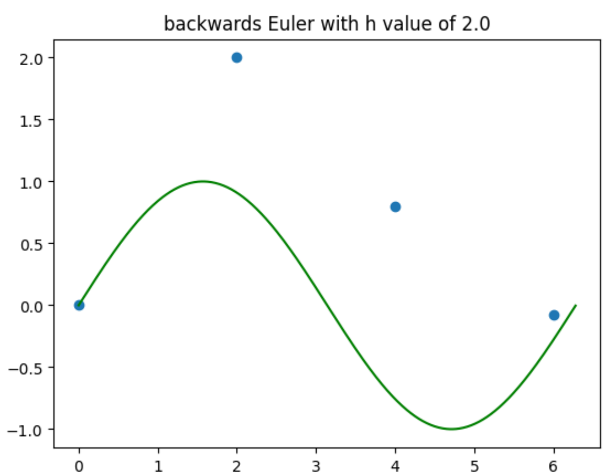
c)











The first trend that I notice is that as the h value increases, the average distance of a point from the actual solution differs more and more. The approximations also tend to bias towards positive infinity to the point where all of the approximations in the graphs of h=1.0 and h=2.0 are greater than the actual solution.

d)

a)

$$y''(x) \to \frac{\frac{y_f(nh+2h) - y_f(nh+h)}{h} - \frac{y_f(nh+h) - y_f(nh)}{h}}{h}$$

$$-y_f(nh) = \frac{y_f(nh+2h) - 2y_f(nh+h) + y_f(nh)}{h^2}$$

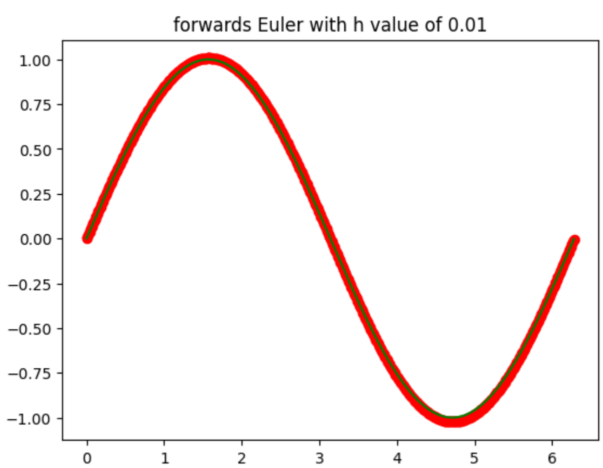
$$h^2 y_f(nh) = -y_f(nh+2h) + 2y_f(nh+h) - y_f(nh)$$

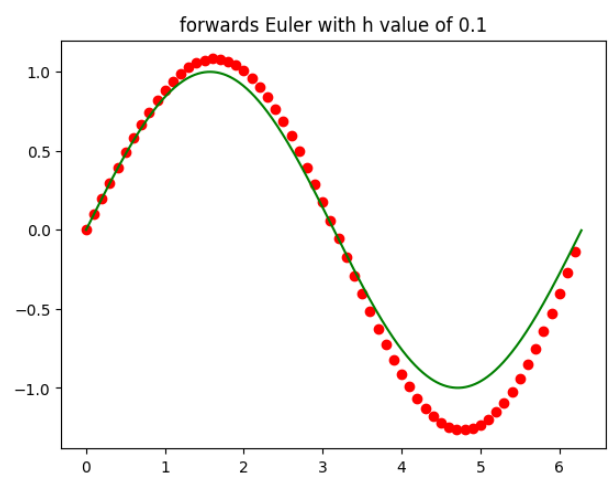
$$y_f(nh+2h) = 2y_f(nh+h) - y_f(nh) - h^2 y_f(nh)$$

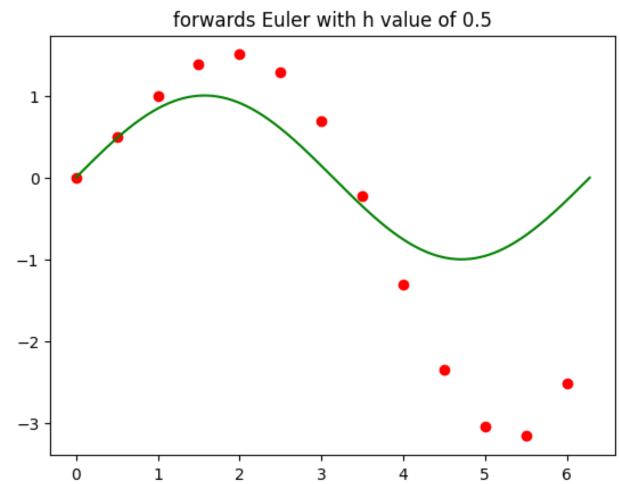
$$\boxed{y_f(nh+2h) = 2y_f(nh+h) - (h^2+1)\, y_f(nh)}$$

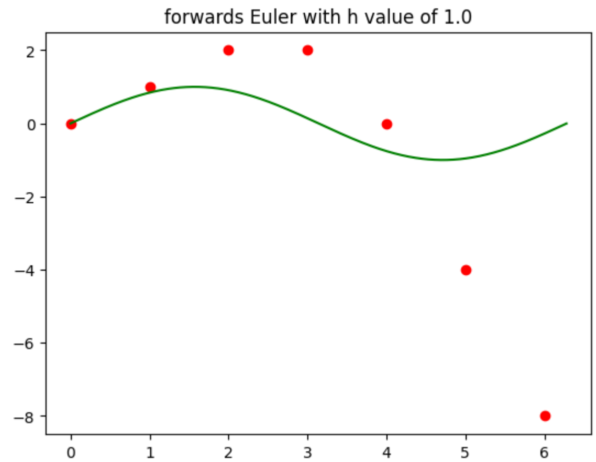
$y_f(0h) = 0$ (same initial points)

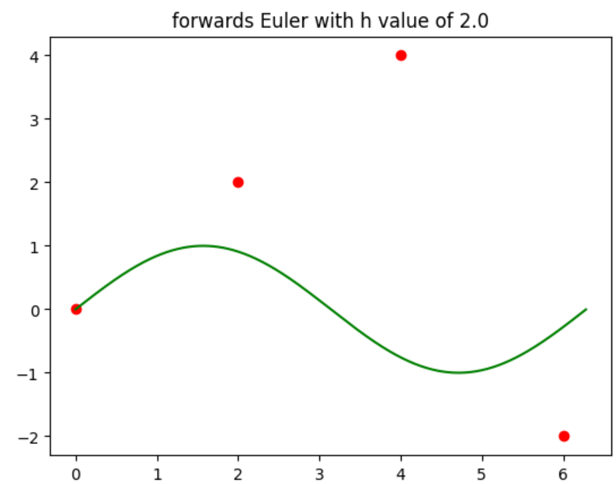
$y_f(h) = h$







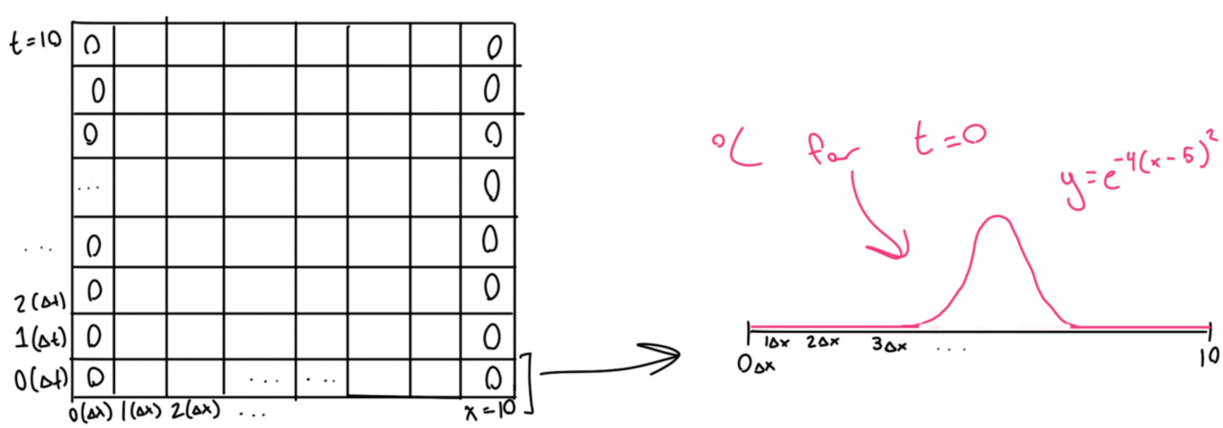




The trends for forwards Euler seem to be reversed in a way. Where the graph of h=0.1 has the approximations biased towards 0, forwards euler has the approximations on the outside of the sinusoid. The part of the curve on the negative side of 0 is also farther from the actual solution than the positive side, which is opposite from backwards Euler. The approximation for h=2.0 also has nearly double the maximum of backwards Euler

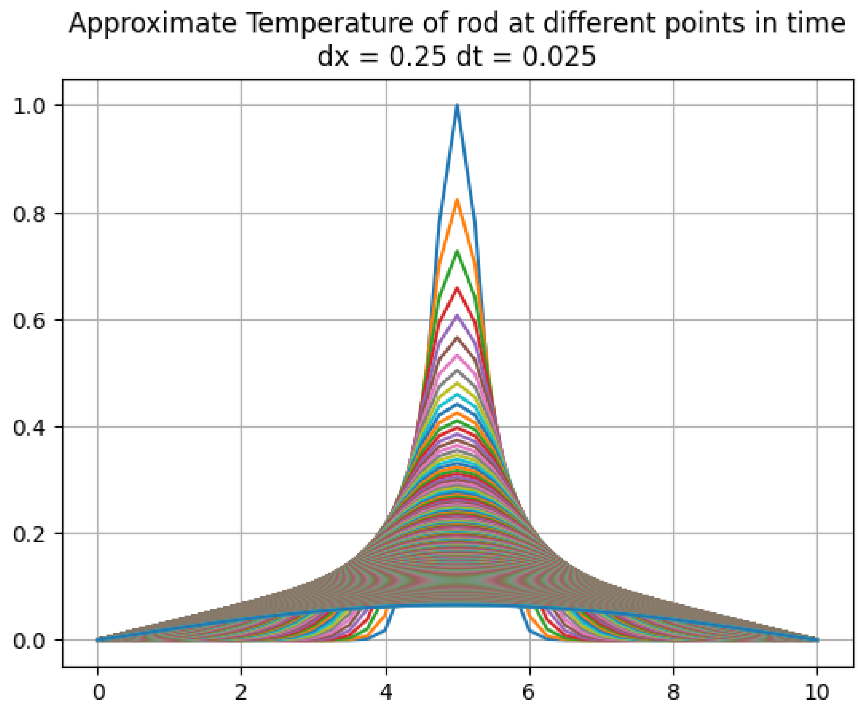
3.

a)



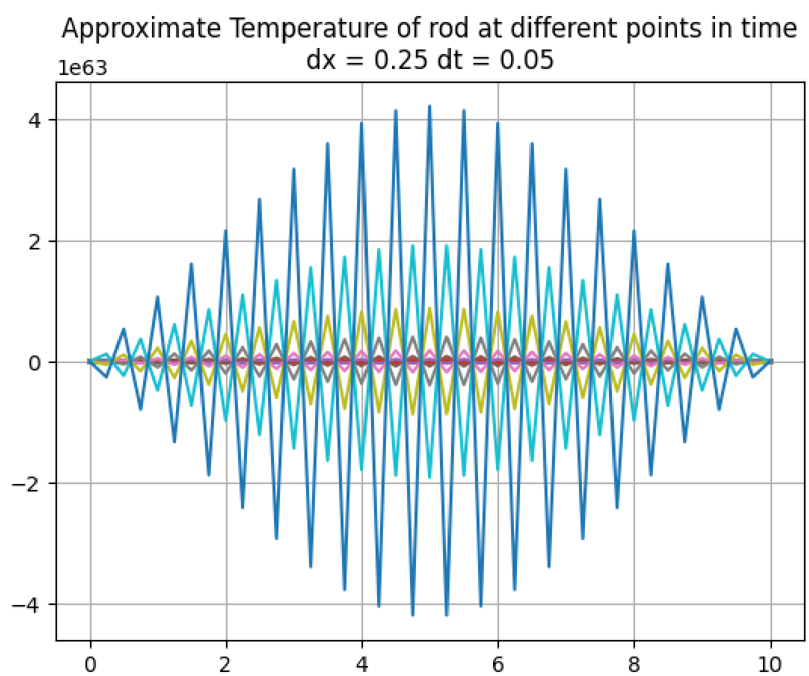
Note: sketching the curve was a little difficult, the actual solution is a bit skinnier/pointy, also I forgot to label the maximum of the curve as 1.0

b)



This definitely aligns with my intuition. The middle of the rod starts the hottest so it will always be the hottest point when no heat is being added. The heat will also dissipate through the rod outwardly so the points that started hot will cool down and the points closer to the outside of the rod will get slightly warmer.

c)



This graph is very obviously wrong. The values quickly become nonsensical and sporadic with the higher value of dt. Other than the obvious zig-zagging, the maximum temperature of a spot on the rod for this graph says it's just over 4 degrees when the maximum starting temperature of a point on the rod was only one degree.

4)

a)

Temporal locality is the idea that an event that just occurred is likely to occur again in a short time. Spatial locality is the idea that an event is likely succeeded by another event that is close or related to that initial event. In terms of memory references, we can consider memory accesses as these events and apply temporal locality to the idea that a piece of memory that was just accessed will likely be accessed again soon, and apply spatial locality as the memory surrounding a piece of memory that was just accessed will likely be accessed soon.

b)

Cold misses are cache misses that occur because it is the first time the data is being accessed. Because the data has not been accessed yet, it is not in cache and must be brought in. Capacity misses are misses that occur because we accessed many other pieces of memory since the last access of the specific piece we want, resulting in our cache filling up (reaching capacity) and evicting the data. Conflict misses are misses caused by our memory getting evicted from the cache by competing blocks that are mapped to the same set as our data piece.

c)

Direct-mapped caches are caches that have just one way. This means that every set in the cache can only hold one block/line. These are the cheapest for locating memory in the cache because we only need to make one comparison because the data we want could only be in one place. A set-associative cache is a cache that has more than one block/line per set. Because these caches can hold more than one block per set, they have the benefit of fewer conflict misses than a direct-mapped cache while still being less expensive than a fully-associative cache. When checking for a hit/miss, we only need to make at most as many comparisons as ways in the cache (blocks per set).

d)

The set size doesn’t have to be a power of 2 because we’re not using any bits in the address to index to our block within the set (like we do in each way with the set #). The set size just tells us how many ways we need to index into using the set# to compare our tag with, which could be a power of 2 or any random number of comparisons like 5 or 7.

5.
a)
line_num = address >> 2
(logical shift)

This essentially chops off the offset bits, isolating the line# bits

b)
cache_index = (line_num << 28) >> 28
(logical shift)

This isolates the bottom two bits of the line number (bits 2 and 3 of the address)

c)
0:
Cold miss. This is the first time this line is being accessed. It loads in line #0 into set #0.
2:
Hit. Tag matches set #0's tag. This block (line #0) is currently loaded into set #0 of the cache.
4:
Cold miss. This is the first time this line is being accessed. It loads in line #1 into set #1.
16:
Cold miss. This is the first time this line is being accessed. It loads in line #4 into set#0
3:
Conflict miss. This address's line was evicted when we loaded address 16's line into set #0. It loads in line #0 into set #0
33:
Cold miss. This is the first time this line is being accessed. It loads in line #8 into set #0.
37:
Cold miss. This is the first time this line is being accessed. It loads in line #9 into set #1.
41:
Cold miss. This is the first time this line is being accessed. It loads in line #10 into set #2.
45:
Cold miss. This is the first time this line is being accessed. It loads in line #11 into set #3.
35:
Hit. Tag matches set #0's tag. This block (line #8) is currently loaded into set #0 of the cache.